Theses and Dissertations 1. Thesis and Dissertation Collection, all items

2006-09

Commercial off The Shelf (COTS) security issues and approaches

Doan, Dung.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/2599

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

COMMERCIAL OFF THE SHELF (COTS) SECURITY ISSUES AND APPROACHES

by

Dung Doan

September 2006

Thesis Advisor:	John Osmundson
Second Reader:	Michael Kochmann

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2006	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE:
Commercial Off the Shelf (Cots) Security Issues and Approaches

5. FUNDING NUMBERS

6. AUTHOR(S) Dung Doan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Custom built products do not always meet the new Department of Defense (DoD) requirements. Their high cost and lengthy development cycle does not suit the Information Age where communication information and technology develop rapidly. To adapt to this new environment, Commercial off The Shelf (COTS) software products have become the core for Military systems. This is the only way to approach the readiness requirements for armed forces. Like any other products, COTS software brings a lot of advantages but also carries side effects for military systems. One of the most serious issues for military use of a COTS software product is its security. This thesis provides an analysis of approaches to identify the security vulnerabilities and recommends an acquisition approach to minimize the issue. It is not the intent of this thesis to find a universal approach to solve the security issue of COTS products.

14. SUBJECT TERMS
COTS Security, DoD Security, Software Security

15. NUMBER OF PAGES
61

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

COMMERCIAL OFF THE SHELF (COTS) SECURITY ISSUES AND APPROACHES

Dung Doan
Computer Engineer, University of Central Florida, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2006

Author: Dung Doan

Approved by: John Osmundson, Ph.D.
Thesis Advisor

Mr. Michael Kochmann
Second Reader

David H. Olwell, Ph.D.
Chair, Systems Engineering Department

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Custom built products do not always meet the new Department of Defense (DoD) requirements anymore. Their high cost and lengthy development cycle does not suit the Information Age where communication information and technology grow fast. To adapt to this new environment, Commercial off The Shelf (COTS) products have become the core for Military systems. This is the only way to approach the readiness requirements for armed forces. Like any other products, COTS brings a lot of advantages but also carries a couple side effects for military systems. One of the most concerning for military use of a COTS product is its security. This thesis will provide analysis of approaches to identify the security vulnerabilities and recommended acquisition to minimize the issue. It is not the intent of this thesis to find a universal approach to solve the security issue of COTS products.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

C/NDI	Commercial or non-developmental items
CC	Common Criteria
CIO	Chief Information Officer
COE	Common Operating Environment
COTS	Commercial Off-The-Shelf
COTS-SPEC	COTS Specification
DAA	Designated Approving Authority
DIACAP	DoD Information Assurance Certification and Accreditation Process
DII	Defense Information Infrastructure
DISCAP	DoD Information Technology Security Certification and Accreditation Process
DoD	Department of Defense
DoS	Denial of Service
DT&E	Development Test & Evaluation
EAL	Evaluation Assurance Level
FAR	Federal Acquisition Regulations
GAO	Government Accounting Office
IA	Information Assurance
IIS	Internet Information Services
IOT&E	Initial Operational Test and Evaluation
JTA	Joint Technical Architecture
MIL-SPEC	Military Specification
NIAP	National Information Assurance Partnership
NSA	National Security Agency
OMB	Office of Management and Budget
R&D	Research and Development
RFP	Request for Proposal

SOW	Statement of Work
SSAA	System Security Authorization Agreement
U.S.	United States
UDP	User Datagram Protocol

ACKNOWLEDGMENTS

I would like to send special thanks to my advisors, Professor Osmundson and Mr. Kochmann, for their guidance, support, and patience during the work in performing this thesis.

I would like to thank all my professors, instructors, and administrators from the PD21-SEM program for instructing me and helping me complete the program.

I would like to thank PEO STRI and the Training Office for giving me the opportunity to gain the knowledge from the PD21-SEM program.

I would like to send thanks to PM CATT and special thanks to Mike Kochmann, John Foster, Sandy Veautour and Rob Miller for giving me this opportunity and allowing me to have flexible time during the program.

I send special thanks to my family and my friends in supporting me and enabling me to finish the program

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Over the past two decades, software technology has become the main part of the solution for military weaponry systems and other applications. Software provides advanced functionality and helps to intelligently equip our forces to fight against any enemies and terrorists. Software represents a significant portion of the defense program's operational capacity and a significant portion of the government's investment. According to the March 2004 report of General Accounting Office (GAO), the Department of Defense (DoD) spent $21 billion on software, representing 40 percent of the DoD Research, Development, Test, and Evaluation budget for fiscal year 2003 (GAO report, March 2004). An estimated of 20% of the total DoD budget will be spent on software by 2007(GAO report, March 2004).

Traditionally, military systems were developed based on customized software. The DoD bought only products that meet military specifications, also known as mil-spec products. This method of software acquisition had a significant impact on government budgets due to immature technology and schedule slippage. During the period of 1994 through 1997, Mr. William Perry, Secretary of Defense, mandated a policy to reduce the dependency on expensive military specification products (Eland, 2002). To implement this policy, the DoD committed to use commercial off-the-shelf (COTS) products rather than buying mil-spec products. COTS software has a lot of advantages as it incorporates newer technology and newer standards, and can be updated faster than custom-built software. Maintenance cost is substantially reduced since COTS software is widely used by a large population. However, COTS software has some disadvantages. Since COTS software's main purpose is to serve popular needs, it ignores some DoD requirements, such as inter-operability, robustness, and security.

This thesis will identify the issues associated with the security of COTS software that are used in military systems. The recommended approaches and techniques to alleviate the issues will be discussed. It is not the intent of this thesis to find a universal approach to resolve the security issues of COTS software, but rather to seek ways to minimize them.

This thesis is organized into the following chapters:

- Chapter II identifies the issues of COTS software used in DoD applications and systems.
- Chapter III identifies the security requirements of COTS software for DoD systems.
- Chapter IV proposes three options for meeting DoD security requirements with COTS software and provides an analysis for each option.
- Chapter V proposes some recommendations for both DoD acquisition and industry to increase the security requirements for COTS software.
- Chapter VI presents a conclusion from the analyses.

II. BACKGROUND

A. WHAT IS COTS?

COTS is the acronym for Commercial Off The Shelf. COTS products are described as either software or hardware products that are commercially available for sale, for lease, or for license to the general public. They are defined by the market's needs, not individual requests. COTS systems can be composed of many other COTS software or hardware components.

Due to the scope of this thesis, only COTS software will be discussed. COTS software has with the following characteristics:

- The source code is in the form of a "black box" because it is delivered in binary code which protects a vendor's proprietary design. Binary code is impossible to reconvert to source code.
- It is significant in functionality and very complex.
- It has periodic releases with newly added features, error fixes, or upgrades for technology advancement.
- The general nature of COTS software design is not to interoperate with other COTS products.

With all of these characteristics, the DoD wants to use COTS as part of either their new developments or upgrades to legacy applications.

B. WHY IS COTS SOFTWARE A REQUIREMENT IN THE DOD SYSTEMS?

Software is a crucial part of our everyday life. It becomes a core part of our basic daily activities that we do not even notice (Main, October 2005). It allows us to make phone calls, check movie show times on-line, access email, and operate cars, as examples. Every single button we push or number we dial from our telephone is controlled by software. It allows the passing of information or data from one end of the telephone network to another. Movie show times are available on the Internet through different websites. Business organizations allow their employees to access work email from home and other remote locations. A preprogrammed computer controls most parts of a car. Airplanes are constructed with millions of lines of code. All of these activities are supported by industrial software. In other words, software development and its acquisition are driven by industry. Human demands increase with higher expectations and a more advanced society. In order to meet society's needs, software development is changing more rapidly than ever before. The development trends are highly dependent on technology, the Internet, and its network. As these change, so does software development. This requires the commercial marketplace to be in the driver's seat of technology advancement, not the DoD. With these changes, there is tremendous opportunity for new commercial systems to implement unprecedented capabilities. The commercial software needs to stay abreast of the technology, needs to apply the technology, and needs to integrate those technologies in a timely manner for commercial needs. This trend also affects DoD acquisitions. Military systems have to be more flexible, scalable, and configurable so that they can satisfy these changing needs. Therefore, demands and requirements for more robust, faster, more integrate-able, and more user-friendly systems are increasing. The intent is to satisfy these requirements with COTS software solutions. In fact, in 1994, William Perry, Secretary of Defense, was first to initiate the mandated use of commercial products and practices in DoD systems and applications (Eland, 2002). The use of commercial off-the-shelf software became the basis for core information technology needs and requirements for the military.

COTS software is expected to be very beneficial to DoD acquisition. The most important advantage of using COTS product is to satisfy the requirement to be cost effective. It is often cheaper to buy commercial pre-existing software rather than to build the same component from scratch or to have it custom built. License fees and performance are predictable. Maintenance cost can be divided equally over the number of customers and licenses and is, therefore, very inexpensive. The extra time to test the software is not required for COTS software during the development phase.

Another advantage is that it has a faster procurement process since it can be reused in other applications. In addition, from the technical perspective, COTS software usually accommodates the latest mature technology and provides rich functionality products.

Recognizing the advantages of COTS products, the DoD has released official policies to guide its acquisition in using COTS products for different types of military systems to maximize benefits. The following documents have been found to guide the use of COTS products in DoD:

The Clinger - Cohen Act was formally published in February 1996 but it did not take effect until August 1996. Its guidance is to incorporate commercial technology for "equipment or systems of equipment used in automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by the executive agency" (DoD, February 1996).

Franklin Raines, Office of Management and Budget (OMB) Director, in his memorandum provided guidance for federal information systems as "support work processes that have been simplified or otherwise redesigned to reduce costs, improve effectiveness, and make maximum use of commercial off-the-shelf (COTS) technology" (Raines, 1996).

Federal Acquisition Regulations (FAR), part 12 stated, "acquisitions should purchase commercial or non-developmental items (C/NDI) when they are available and they meet the needs of the organization" (GSA)

DoD Regulation 5000.2 "exploits and reuse COTS" and "consider C/NDI [to be] the primary source of supply" (DODD 5000.2).

Joint Technical Architecture (JTA) and Defense Information Infrastructure (DII) Common Operating Environment (COE) guidance are essentially based on commercial standards and products.

C. ISSUES WITH COTS IN DOD APPLICATIONS

Unfortunately, using COTS products for military applications has some drawbacks. First, security is the greatest concern for the military due to the characteristics of COTS. COTS software is consumed in the form of a "black box" or "as-it-is". There is no security warranty. With COTS, functionality takes higher priority than security. In other words, security is usually at the bottom of the list. COTS software is widely available which increases the risk that it falls into the hands of users with malicious intent. These users have the potential to uncover security flaws, which then puts the software and systems built on the software at risk for attack. COTS software is large and complex with millions of lines of source code (Lipson, Mead, and Moore); hence, program bugs can easily lead to security vulnerabilities.

Configuration control is also a nightmare. COTS vendors constantly update their software to apply new technology and to meet the majority of their customers' demands. Upgrades usually require other updated components from multiple vendors and usually upgrades are not backward compatible with the rest of the system.
In this thesis, only security issues will be discussed.

1. Historic Events of Unsecured Software in DoD Programs

In highly reliable DoD systems, the defect of any function or software component may result in tragedy and cost. Let us review some history of events:

- In December 1994, hackers attacked the U.S. Naval Academy's computer systems and deleted the master backup file from one system. It blocked access for authorized users to another system and tampered with twelve thousand passwords and compromised a main router, the electronic equivalent of a computer system's arteries. According to the General Accounting Office report in May 1996, "the attacks caused considerable disruptions to the Academy's ability to process and store sensitive information" (Browning).

- Hackers attacked Rome Laboratory, the Air Force's premier command and control research facility in New York, in March and April 1994. The attackers used Trojan horses and sniffers to access and control Rome's operational network. Hackers took over control of Rome's support systems for several days, copied and downloaded critical information such as air tasking order systems' data. The estimate of attack's cost was over $500,000 which included time to take systems off the networks, verify systems' integrity, install security patches, restore service, and costs incurred by the Air Force's Office of Special Investigations and

Information Warfare Center. The cost incurred from losing data was not included because the attackers were never arrested (Slabodkin).

- During 1995 and 1996, an Argentinean hacker used Internet connections to break into computers at the Naval Research Laboratory, NASA, Los Alamos National Laboratory, and other Defense Department sites. According to the GAO, the systems contained "sensitive research information, such as aircraft design, radar technology, and satellite engineering that is ultimately used in weapons and command and control systems" (GAO Report, 1996).
- “Dutch hackers who pillaged computer files at thirty four U.S. military sites in the months leading up to the Gulf War offered the information to Iraqi leaders, a former Energy Department official said in March. The hackers not only learned the exact locations of U.S. troops and the types of weapons they had, but also gained information about the capability of the Patriot missile and the movement of American warships, Eugene Schultz, the former head of computer security at Energy, told a London newspaper” (Browning).
- “A hacker who broke into and defaced the Air Force site on the World Wide Web, the multimedia corner of the Internet, late in December forced the temporary closure of eighty Web sites that carried, among other things, information on Gulf War illnesses” (Browning).
- In September of 1997, the Yorktown's propulsion system failed. The ship had to be towed to a Navy base at Norfolk, and the ship was not restored to operational status for two days. The culprit? The software running on PCs designed to control the ship crashed, taking the rest of the ship down with it. The Navy had a policy of using commercial off-the-shelf hardware and software. The Smart Ship program used standard Pentium Pro PCs and the standard operating system Windows NT 4.0. It turned out

the software that was used for Yorktown had bugs including the operating system. NT 4.0 was unreliable and caused the failure (Carhart, Chan, and Hu, 2000).

- “In 1998, the accidental failure of the Galaxy IV satellite disrupted over thirty five million pagers across the United States for two to four days, and blocked credit card authorization of point of sale terminals” (Poulsen).
- In January 2001, the attack of Code Red, Nimda, one of the most costly assaults launched across the Internet in fifteen minutes took down three hundred thousand Internet servers, 911 emergency phone service, ATM machines across the US, and a major airliner’s automated reservation system. This attack resulted from the security vulnerabilities in Microsoft's Internet Information Services (IIS) Web server product (Costello).
- In August 2004, hackers took down the weaknees.com, amazon.com, and the Department of Homeland Security Web sites. The attacks caused more than $2 M in damage (Krebs).

According to a report from the Treasury Department's Office in December 2005, the cost of cyber crime in 2004 went up to $105 B. Also, the military spends about two billion dollars each year on information assurance on encryption systems and Research and Development (R&D) for new capabilities (Wait).

2. Types of COTS Software Security

Security vulnerabilities are the results of software weaknesses, faults, logical errors, and flaws. Depending on how COTS software components are used in DoD system, their misbehaviors can breach the security via the following methods:

- The component might allow accessing unauthorized resources or services of the system (Zhong and Edward).
- The component might allow accessing a resource in an unauthorized way that causes another component to fail its functions in the system (Zhong and Edward).

- The component might abuse authorized privileges and take control over the system or other components of the system (Zhong and Edward).

The DoD has taken steps and released policies in an effort to reduce the cost of and to minimize the number of security breaches. This will be discussed in further detail in the following chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

III. COTS SOFTWARE SECURITY REQUIREMENTS

Knowing that COTS software will never meet DoD security demands and realizing the damages, cost, and risks if deploying unsecured COTS software for military applications, DoD still wants to use it to fill the core information technology needs for the military because of other advantages such as advanced and mature technology, timing, and readiness.

The security requirement is clearly stated in DODI 8500.1, "all DoD information systems shall maintain an appropriate level of confidentiality, integrity, authentication, non-repudiation, and availability that reflect a balance among the importance and sensitivity of the information and information assets; documented threats and vulnerabilities; the trustworthiness of users and interconnecting systems" (DODD 8500.1) and "all military voice radio systems, to include cellular and commercial services, shall be protected consistent with the classification or sensitivity of the information transmitted on the system" (DODD 8500.1).

In the Defense Acquisition Guidebook, paragraph 9.3.1, requirement for security testing "commercial items, regardless of the manner of procurement, undergo DT&E to verify readiness to enter IOT&E, where operational effectiveness, suitability, and survivability for the intended military application are demonstrated. Programs should not enter IOT&E unless the DoD Components are confident of success" (DODD 5000).

Unfortunately, DoD policies and guidance are not currently addressing how to protect software. It focuses primarily on operational threats, not insider threats such as the insertion of malicious code by software developers.

The next chapter will propose three techniques to improve security for COTS software when they are used to develop or upgrade DoD systems.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. TECHNICAL APPROACHES AND ANALYSIS

To reduce the vulnerability of COTS software, there are currently three approaches recommended by the Chief Scientist of the Cost Research Department at PRICE systems, Arlene Minkiewicz (Minkiewicz).

Each approach will be discussed in detail with techniques, advantages, and disadvantages.

A. BUYING COTS AND WRAPPERS

1. What is a Wrapper?

Software wrapper is considered as a shell or a box that isolates a component from other components and its processing environment. It is responsible for connecting information sources to external components (Figure 1).

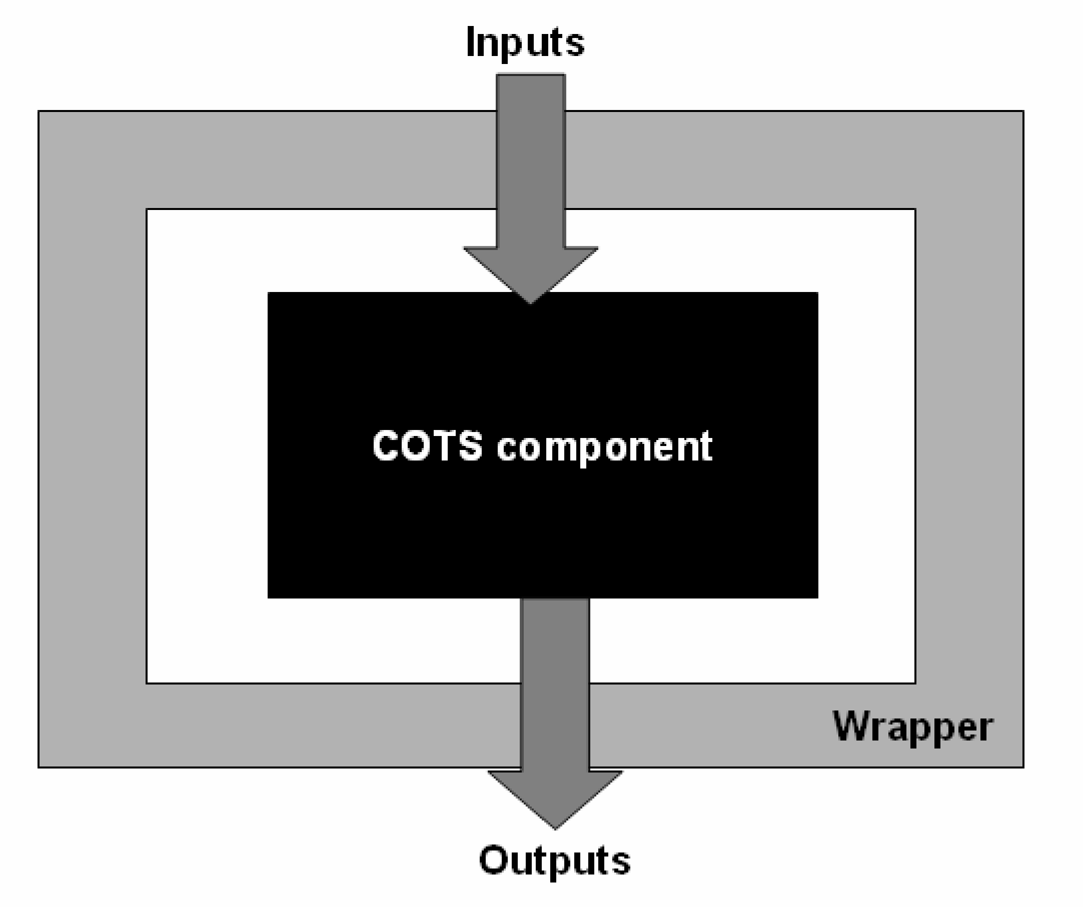

Figure 1. Wrapper Technique

In terms of the security wrapper, it provides necessary transformations and filtrations of data that is coming in and out of the wrapped components. It disallows unwanted information to enter the component by placing a software barrier around the component. It also controls the outputs of the component.

2. Buy-and-Wrap Approach

In this option, the DoD agency or project manager plans to purchase the COTS components that best meet the functional requirements for the system. Then, the project manager will have developers build security wrappers to secure these purchased components.

There are two different methods that can be used to develop security wrappers.

The first method is to keep certain inputs from reaching the component. Wrappers filter inputs before they are called from the COTS component. Wrappers only allow the call from the COTS component if the input is likely to produce acceptable behavior. This is the way to prevent the execution of improper inputs, and this may limit the component's output range.

The second method of incorporating a wrapper is to capture the outputs before the component releases them. Outputs will be checked to ensure they meet certain constraints and then the wrapper will qualify the outputs before release.

Both methods can be used at the same time to minimize the vulnerabilities of COTS components.

3. Advantages and Disadvantages

Using wrappers does not mean that the component is free of vulnerabilities. Unwanted outputs sometimes bypass a wrapper due to the design of the wrapper. For example, if it is not known that a component can call the operating system to delete a file, the wrapper probably won't be designed to prevent it. If, however, it is known that the component needs to delete temporary files that it creates, the wrapper can be designed to allow only file delete requests pertaining to those temporary files. The value of wrappers, then, depends on how well they are designed. They cannot protect against events unanticipated by developers—such as those events caused by Trojan horses.

This approach is both complex and prone to errors. The significant problem is that the system developer must write code that gracefully handles inputs that cannot properly be handled by the COTS component. Designers should ensure that the wrapper is correctly filtering unwanted inputs and outputs. This is extremely difficult since the COTS components are delivered in a “black box” and in binary executable form. Furthermore, wrappers are unable to track all temporary files (outputs) that are created during and after the wrapping operation. A good example is the Microsoft Word application. The undo feature allows the creation of a temporary buffer file where data is immediately stored. This way, the application can let the operator perform the undo and redo functions. Wrappers have to capture all of the temporary files in order to ensure the component is a hundred percent secured.

There are advantages for this approach. First, there are a variety of COTS vendors who provide the components that can meet the DoD requirements. This results in buying better and cheaper products. The second advantage is maintenance. The level of reuse of wrappers can be maximized. The modification of a security wrapper is a minimal effort if one COTS component is replaced with another. This approach also fits well into the long and painful process of DoD certification procedures. The last advantage of this approach is that it will be independent of the wrapper updates if the inputs and outputs of the component remain unchanged.

B. BUYING ONLY PRE-CERTIFIED COMPONENTS

Before getting into the approach, we need to understand the definition of Common Criteria.

1. What is Common Criteria?

Common Criteria is a process to evaluate the reliability of computer products. It is recognized as an international standard. It uses a numerical rating to reflect the assurance requirements, Evaluation Assurance Level (EAL), of a computer product. There are seven EAL levels which range from EAL1 to EAL7 (GAO Report, March 2006). The higher the evaluation level is, the more secure the product is. Depending on the level of security required for the system, COTS can be certified at one of the seven levels. EAL1 is the lowest level where security functions are certified by analysis.

Certified products at this level provide some operational confidence but are weak on threats. Usually, EAL1 is used only if damages to a product cause minimal damage to the agency and the risk of threats is also very low. An example of this would be if the threat only appeals to amateur hackers. To certify a product at EAL1, a vendor only needs to provide very basic documentation for analysis. The vendor also has to provide useful protection against identified threats. For DoD applications, any threat or damage to a DoD application might cause result in loss of life or loss of millions of dollars.

EAL2 is relevant to products that may incur some, yet a small amount of damage to consumers when products fail (GAO Report, March 2006).

EAL3 is applicable to products that may allow a few instances of serious damage if the application fails (GAO Report, March 2006).

EAL4 is for products that may allow serious to extreme damages if the product fails. Also, these products have a high risk of attack (GAO Report, March 2006).

EAL5 is applied to high risk products. In this case, damage caused by the failing security requirement may lead to serious financial or a breach in infrastructure of the consumer (GAO Report, March 2006).

EAL6 is applied when a product has a high risk of being attacked. Currently, there is no application that is certified at the EAL6 level or higher (GAO Report, March 2006).

EAL7 is applied to products that bring a very high risk of being attacked. These types of products may incur damages where the failing products are significant to many consumers (GAO Report, March 2006).

2. Approach and Technique

Currently, COTS products can be certified for security based on EAL by the National Information Assurance Partnership (NIAP) which is a Government agency and is managed by National Security Agency (NSA).

The principle of this approach is that the DoD agency purchases the COTS products that are certified or stamped with the required level of DoD security level (Minkiewicz).

The process is very simple for COTS vendors who want to certify their products and is depicted in Figure 2. First, vendors will submit product documentation and the COTS software to the NAIP laboratory. Documents have to include a protection profile and security target for the product. NAIP lab will then go through its internal process of evaluation, test, and validation. NAIP then issues the report. If the product passes the common criteria of the requested level, NAIP then issues the certification and publishes the product in the validated product list. A DoD project then can select a certified COTS product for its development and integration.

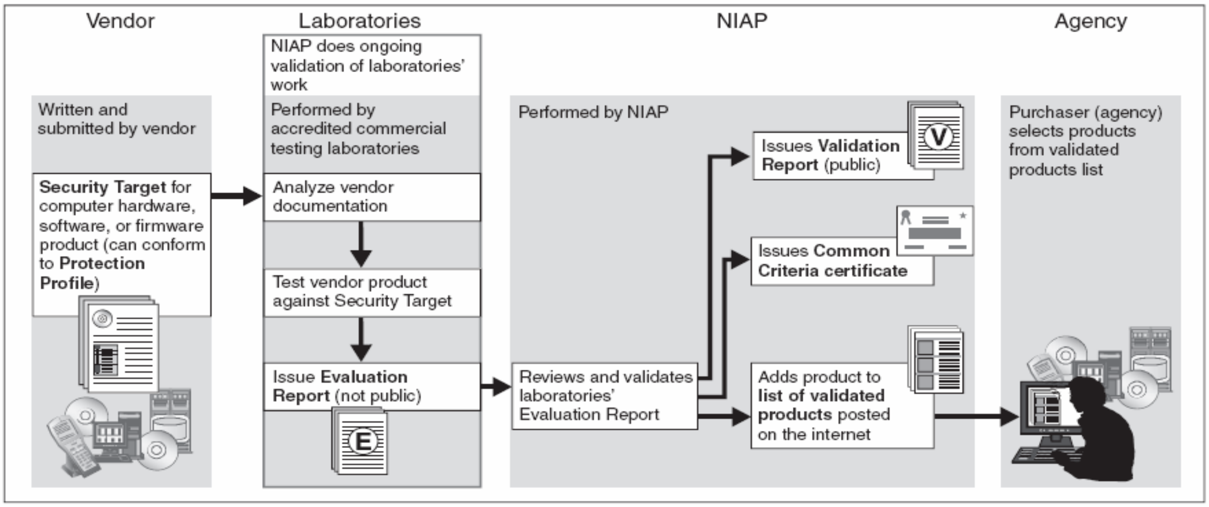
Vendor
Laboratories
NIAP
Agency
NIAP does ongoing validation of laboratories' work
Written and submitted by vendor
Performed by accredited commercial testing laboratories
Performed by NIAP
Purchaser (agency) selects products from validated products list
Security Target for computer hardware, software, or firmware product (can conform to Protection Profile)
Analyze vendor documentation
Test vendor product against Security Target
Issue Evaluation Report (not public)
E
Issues Validation Report (public)
V
Issues Common Criteria certificate
Reviews and validates laboratories' Evaluation Report
Adds product to list of validated products posted on the internet

Figure 2. Certification process of COTS product through NAIP (GAO Report, March 2006)

There are two options to certifying upgraded products. One option is to validate only the updated portion. The other option is to send the whole product through the process. The option is up to the software vendor.

The fee and the time it takes to certify vary. For example, it takes from four to nine months to have software certified at EAL2. The fee is in the range from $80,000 to

$200,000. For EAL4, it takes up to 24 months and the fee is up to $350,000. The higher the EAL level the longer it takes, and the higher cost. Prices to certify at EAL6 and EAL7 have not been released (GAO Report, March 2006).

3. Advantages and Disadvantages

As an advantage, this approach allows the products to be used with some confidence levels. It might also reduce the integration effort in trying to develop protection for COTS software components.

This approach has a limitation in that vendors are put in a position to satisfy the system's security requirements. In addition, only certified COTS components will be evaluated as candidates for use. This prolongs the development time and increases the integration effort since it limits the selection. It may also lead to the necessity to develop more in-house functionality if properly certified components are unavailable for some functional requirements.

The certifications seem to be somewhat wasteful. The Common Criteria certification is not necessarily associated with strong and secure products. Rather, EALs refer to the level of confidence of the evaluation; they are not the level of security the product provides. We can use Microsoft Windows as an example. It is certified at EAL4, but the product is not secure. Vulnerabilities appear on a regular basis where consumers receive security patches and "hotfixes" monthly and quarterly.

These metrics of certification are hard to quantify due to the limitation of the evaluation environment. When a COTS vendor submits their software, there is a list of assumptions for the product such as inputs, operational environment, operating system, hardware platform, memory, and third party applications such as drivers, and other things. A wide range of tests needs to be run in order to collect enough data and confidently analyze the software. In addition, there is no warranty that the lab or evaluator was able to create every case that consumers might run into that might cause damage to the software.

This approach is likely to increase the purchase and maintenance costs of COTS components. Every update of COTS has to go through the certification process. It takes additional time and costs more money to certify the update.

The certification process is not practical. It sounds like a “pay for play” game that COTS vendors have to sign up for. Any industry who wants to sell products to government organizations is required to be at least EAL2. To be certified at EAL2, the only thing a vendor needs to do is to document and describe what their product actually does.

C. BUYING COTS AND CERTIFYING IN-HOUSE

The principle of this approach is that the DoD agency purchases the COTS software that best meets its requirements and then has these components certified at the DoD required security level. The Project Manager who decides to buy those COTS components will be fully responsible for complying with DoD security requirements and having to go through a DoD certification process.

1. Methods

There are two methods that the purchaser can use to certify their COTS software in-house.

a. Black Box Testing

The fundamental of this technique is to treat the test component as a “black-box”. The principle of black-box testing is to test the functionality of the software (Du). It is similar to testing an electronics part. Consider a light bulb as an example. A consumer buys it, installs it, and turns the switch on. If the light does not light up, the bulb will be removed and returned to the store or the manufacturer. The same approach applies for software. It is tested without any analysis of the code and testers do not need to have any knowledge of the software. They only have to see if it works by flipping switches (inputs) and seeing what happens to the lights and dials (outputs). It is used to test the specifications of the project. A simple black box test approach is described in the Figure 3.

Input
Output

Figure 3. Black Box Testing

The advantage of this approach is that it does not require the understanding of the internal structure of the software; therefore, a low level technical person can run the test, which reduces the expense of training. In some cases, the test can be set up to run automatically. The black box test does not require accessing the vendor's source code (Voas).

However, this method requires that the tester has the knowledge of "legal" inputs and what the expected outputs are for legal inputs. This can be very hard for DoD systems since a certifier has to go through a test case for every allowed input. The test case is another issue in that a tester has to be able to identify every condition of the software for every case of input in order to catch all errors that are potential vulnerabilities for the software. Using a black box test alone to certify a COTS component is insufficient due to the limitation of test cases (Du). A tester might not be able to test all cases or be able to create all test cases that fail the test software component. For example, a software vendor claims that the inputs for software they market can handle 255 characters. Will tester check all cases? This will include over 255 cases that tester has to run. She or he has to test all 255 cases that would produce the expected results. In addition, the tester also needs to run cases outside the set limit. It might be impossible for tester to run all cases with all combinations of test cases.

b. Fault Injection

Fault injection has been used widely for different purposes. It is used to test hardware for tolerances such as finding fault time on hardware (CPU, circuits, electronic parts, etc.). This technique is also used extensively in testing software effectiveness. In the context of this thesis, it is used to test the security vulnerabilities for COTS software. The main goal of the fault injection approach is to insert faults into the COTS software to test the security tolerance (Clark & Pradhan). There are three methods to insert faults. If a fault is injected before the execution of the COTS application, it is called pre-execution fault injection – the first method. The second method is called execution fault injection. Faults are injected when the application is running. The third method is the combination of the first and second ones. This means that faults are injected before and during the application execution (Du). An overview of such methods is shown in Figure 4.

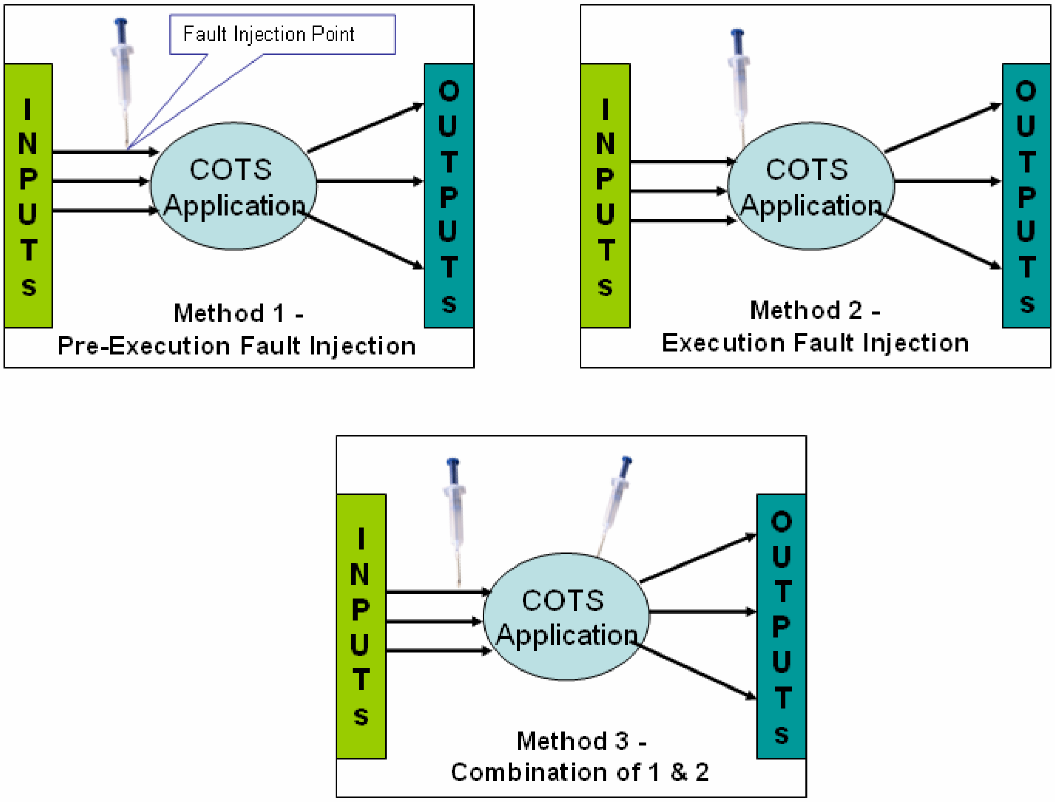

Figure 4. Fault Injection Methods to test security of COTS software (Johansson)

The fault infection method, no matter which method is used, has the following steps (Johansson):

- (1) Finding the fault injection point

- (2) Defining faults

- (3) Injecting the faults

- (4) Analyzing outputs for the vulnerabilities

The challenges for the fault injection approach are in steps 1, 2, and 4. Fault injection allows inserting errors without understanding how the software is structured but it requires understanding the software behavior in order to analyze the outputs. This technique relies on the security reviewer’s knowledge. The result of the analysis is very important in the conclusion of whether the products are vulnerable or not. For example, if the result returns a positive evidence of vulnerability, there are two cases. If the viewer correctly analyzes the result, then the software is in high risk of security breach, or, if the reviewer’s analysis is incorrect, the software might be free of vulnerabilities. Another issue with this approach is that it does not catch all vulnerabilities due to missing defined faults.

On the positive side, this approach can detect systematic anomalies due to the functional anomalies tests. Fault injection can also be done automatically.

2. Advantages and Disadvantages of this Option Approach

This approach allows the detection of systematic anomalies besides security breaches. It increases the confidence level of the product since reviewers analyze outputs in multiple cases. This approach can also be done at some automatic level which reduces the cost in resources. Third, once fault models are developed, they can be reused to test the updated version of the COTS software. This approach allows the wide choice of available vendors that meet other DoD requirements.

However, this approach is a very risky. First, if the certification fails, the buyer must then restart the evaluation and selection process to identify other potential solutions and must contend with the fact that they have wasted effort in purchasing unsecured software that does not meet the requirements (Minkiewicz). Secondly, it takes days and months to come up with the test template as well as developing the fault injection approach. This is not a cost effective approach.

V. RECOMMENDATIONS

As software vulnerabilities are inevitable and as the military moves from yesterday’s MIL-SPEC procurements to COTS-based procurements and strategies, the risk of being vulnerable is getting higher. The military is going from yesterday’s MIL-SPEC procurements to COTS-SPEC procurements to take the advantage of new technology and cost effectiveness, but only if they can adapt to commercially available software. The technologies and methodologies to minimize vulnerabilities of such DoD software systems come to the attention of not only DoD acquisition personnel but also commercial software producers. This chapter will propose recommendations for both the DoD acquisition community and industry to reduce the security risk for COTS software.

Before getting into the recommendation, we first need to look at what are the causes of software vulnerabilities. Secondly, we look at the differences between DoD and commercial acquisition. Lastly, recommendations to reduce the risk of software security vulnerability in DoD and commercial are introduced.

A. CAUSES OF SOFTWARE VULNERABILITIES

There are two main factors that can result in a software breach. The first one is due to poor specification, design, and architecture. The second one relates to programming bugs.

Most of the time, DoD specifications describe security requirements vaguely and poorly. Reviewing some DoD projects, the most common phrase is “Information Assurance (IA) compliance” in the specification and the Statement of Work (SOW). What does it mean? At what secure level does the software have to be in order to meet the “compliance” requirement? Each EAL has its own requirements for security. DoD plans to buy COTS software with vague and weak security requirements. In addition, commercial software vendors only concentrate on cost, schedule, features, and functions requirements most of the time. Security requirements become “nice-to-have” functions. This means that security issues will be left for the maintenance phase of the acquisition. For example, at the buying phase, a COTS product is used as a component for a web application and the requirement of the application only concentrates on the data rate. User Datagram Protocol (UDP) is used in that COTS software to meet the data transferring rate. There is no other requirement such as “guaranteed delivery.” The selection of that COTS product has jeopardized the application with the unreliable protocol, UDP. As a matter of fact, UDP has been implemented into some Trojan horse viruses. Hackers develop scripts and Trojans to run over UDP in order to mask their activities. UDP packets are also used in Denial of Service (DoS) attacks. UDP meets the performance requirement (data rate) but does not satisfy the unwritten security requirements. Many times, specifications do not clearly define the mission-critical and security requirements of the software. This leads to the high risk of security breach.

Besides specifications, programming bugs also constitute security breaches. Programming bugs such as errors, flaws, mistakes, failure, or faults during the code writing process can be created by careless programmers. The common semantic errors includes dividing by zero, infinite loops, arithmetic overflow or underflow, exceeding array bounds, using an uninitialized variable, accessing memory not owned, allowing memory leak, stack overflow, buffer overflow, deadlock, loss of precision in type conversion, and others. The results of software bugs could cause serious security breaches. For example, from the BBC news on 18 March 2003, the loophole error in the Windows 2000 operation caused a security breach and the US Army server was about to be attacked. With this loophole, attackers could have completely taken control over hacked computers and gotten into secure information from other server computers. Fortunately, there were no serious damages since the Windows 2000 patch came out at the right time and the US Army server was not connected to important servers at that time. Let us take a look at another example. Code errors can cause a buffer overrun, which can lead to the exploits of a Blaster worm. Blaster attacks a computer by blocking or restarting the affected computers, and then it rapidly spreads through the network.

B. DIFFERENCES BETWEEN DOD AND COMMERCIAL ACQUISITION

Next, let us take a look at the differences between DoD and commercial acquisition in order to have a better recommendation.

From the procurement perspective, military software requires tremendous time in preparing competitive bids, or request for proposals (RFPs). DoD already spends a significant portion of its budget creating RFPs. There are considerable sets of deliveries such as status reports, financial reports, schedules, meetings, and others. These deliveries are usually expensive which leads to the high cost of the final product. On an average, a military project spends about 50% of its budget on paper work. DoD services are non-profit agencies and are not as motivated to be driven by the "bottom line", so they tend to overspend or develop software in a way that is more costly than the commercial sector. If the fundamental mission requirements are not compromised, there is no reason for them to improve. Military procurement also involves contract law and policies that often result in a schedule delay on an average from six to 18 months before reaching a final procurement decision that allows the contractor to start the work. Military management has the tendency to rush the development phase to accommodate the time spent on developing the RFP. Unfortunately, schedule pressure causes software failures.

From the requirement aspect, military software is function-driven. It is complex since it deals with weapon systems, communications, radars, and other technical systems. Software is created to serve military operations in both peace and wartime. Mission, safety, reliability, and security are important. Military requirements pay attention to the quality of the product. The commercial sector has a more benefit-driven acquisition. On the other hand, COTS software producers are trying to minimize spending, shorten development time, and create the most profitable products. Therefore, quality control is minimal. This explains why COTS software has a lot of bugs.

Regarding effects of a failure, if military software fails, the operation may fail, lives might be lost, a battle might be lost, and national security also may be in jeopardy. In the commercial world, there are no laws that punish software vendors if their product fails. The worst case that can happen is that they may have to refund money at some level.

C. RECOMMENDATIONS FOR DOD AND COMMERCIAL ORGANIZATIONS

To reduce the risk of being breached when commercial software products are embedded in military systems, the software security requirement should be given a top priority and concern.

1. Recommendations for DoD Acquisition

To ensure highly secure COTS software, DoD acquisition should change the old mentality of “all about writing reports and counting those reports" (Holmes).

First of all, DoD has to clearly define security requirements for COTS components as well as the system. DoD acquisition personnel have to create a document that clearly and completely identifies all security requirements and objectives. This document should be created by the DoD acquisition personnel and reviewed by the Chief Information Officer (CIO) of the organization. This document should be similar to the Protection Profile defined in CC standard where it completely describes security objectives, security related functional requirements, and information assurance requirements. This document should be used during the bidding or choosing the best COTS components.

After security requirements are fully defined, a method for choosing secure COTS components in DoD application has to be established. As stated previously in Chapter IV, the approach 2 (Buying Pre-Certified COTS only) with the addition of developing and using fault injection is recommended. Software developers cannot be counted on to determine if the certified COTS component can be claimed as secure. Reality proves that pre-certified COTS products still have the high risk of security breach. The most famous product is Microsoft Windows. Although Window XP was certified as EAL4 in November 2005, patches have been continuously updated. For this reason, fault injection should be developed and used to minimize security breaches. Using the fault injection method will increase the confidence level of the software being vulnerability-free. Even though the development of fault injection is expensive, it is worth it to test for those costly COTS software and for mission-critical applications.

DoD acquisition should carefully analyze software to see if the security requirements of COTS software meet the system application. Budden, senior programs manager at AVISTA in the CrossTalk Journal published in February 2003, recommended the process in Figure 5. (Budden).

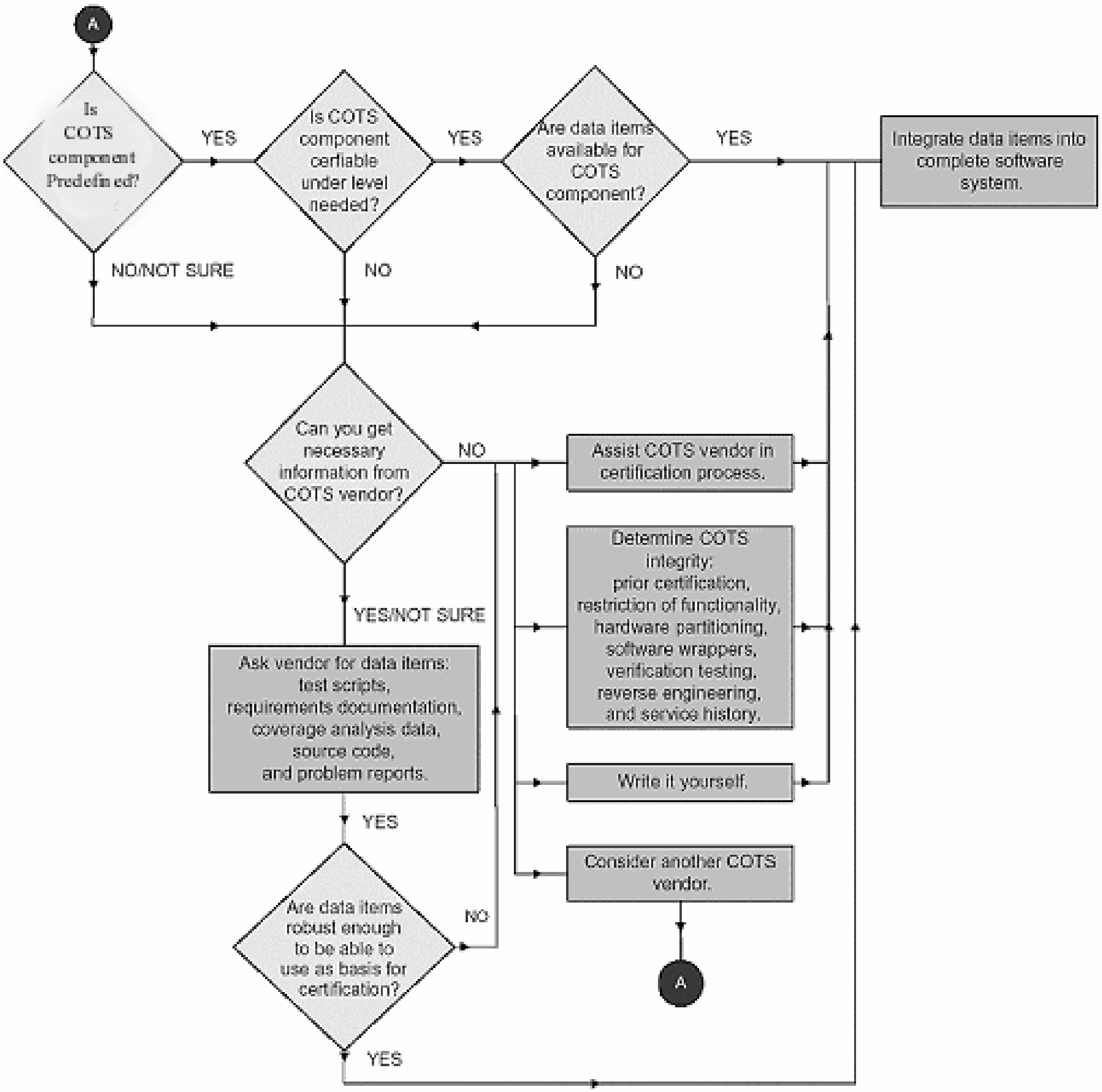
A
Is COTS component Predefin ed?
YES
Is COTS component cerfiable under level needed?
YES
Are data items available for COTS component?
YES
Integrate data items into complete software system.
NO/NOT SURE
NO
NO
Can you get necessary information from COTS vendor?
NO
Assist COTS vendor in certification process.
Determine COTS integrity: prior certification, restriction of functionality, hardware partitioning, software wrappers, verification testing, reverse engineering, and service history.
YES/NOT SURE
Ask vendor for data items: test scripts, requirements documentation, coverage analysis data, source code, and problem reports.
Write it yourself.
YES
Consider another COTS vendor.
Are data items robust enough to be able to use as basis for certification?
NO
A
YES

Figure 5. Process to certify COTS in DoD acquisition (Budden)

It is important to note that standalone COTS certification does not mean that the COTS components are safe for use in the military as a system. It only means that the COTS component has the low risk of a security breach. Certification should be done in the environment that the COTS components operate, not through documentation and other environments.

DoD has its own process to certify and accredit information technology systems. DoD just published DoD Information Assurance Certification and Accreditation Process (DIACAP) in June 2006. This process serves the same purpose as the DoD Information Technology Security Certification and Accreditation Process (DITSCAP) that is to certify & accredit DoD systems for information assurance. The only major differences between the two processes are that DIACAP adds the Net-centricity, Global Information Grid (GIG) and Federal Information Security Management Act (FISMA) concepts. Both processes are very similar in terms of required tasks for certification. Since the scope of the thesis is in the COTS software, concepts of Net-centricity, GIG, and FISMA are not the issue of COTS software, they are more at system level; only the DITSCAP will get introduced. The second reason that the thesis only discusses DISCAP is that DIACAP is new and it has not been widely applied to any DoD systems yet. Details of each task can be found in DODI 5200.40.

DITSCAP composes of four main phases: Definition, Verification, Validation, and Post Accreditation (DODI 5200.40). Phase 1 is to understand the mission, the environment and its architecture to determine the security requirement level. Phase 2 verifies the evolving system's compliance with the System Security Authorization Agreement (SSAA) established process. Phase 3 validates the information set forth in Phase 2 and, finally, Phase 4 starts after the system has been certified and ensures the maintenance of system security. At each phase, specific tasks and documentation are assigned and must be completed to exit the current phase and ready to go to the next phase.

Phase 1 is considered as the definition phase.

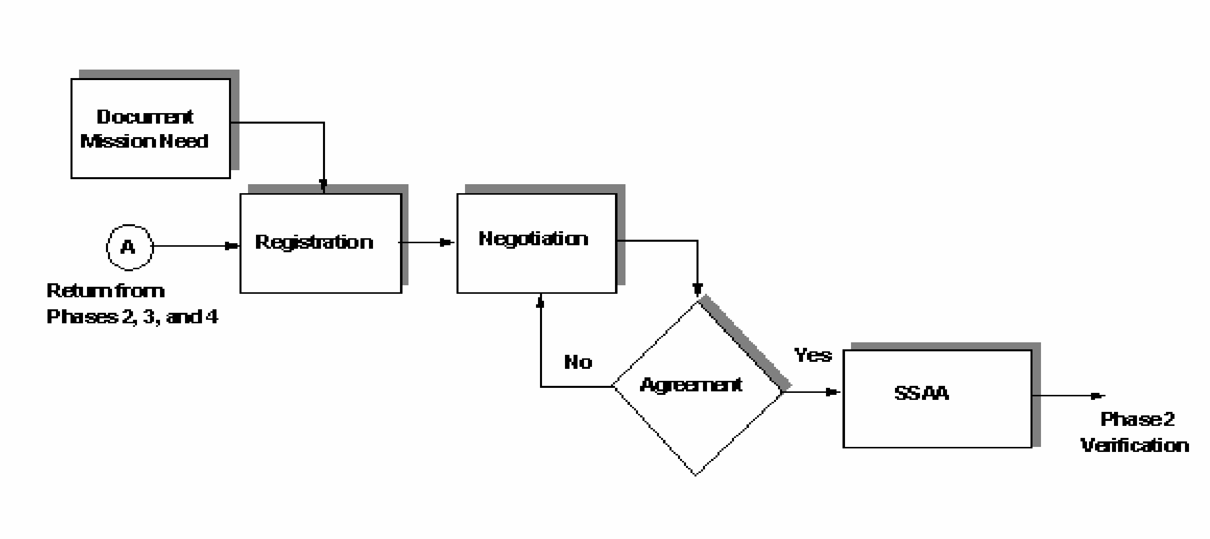

Figure 6. Phase 1 – Definition (DODI 5200.40)

In this phase, the tasks of gathering information about the target system and its environment and agreeing on security requirements, level of effort, schedule, and resources necessary to complete the process have to be executed.

Once these two tasks complete, Phase 2 can start with verifying the target system's compliance with the information agreed upon in Phase 1, identifying and analyzing vulnerabilities, and verifying that appropriate security controls are in place.

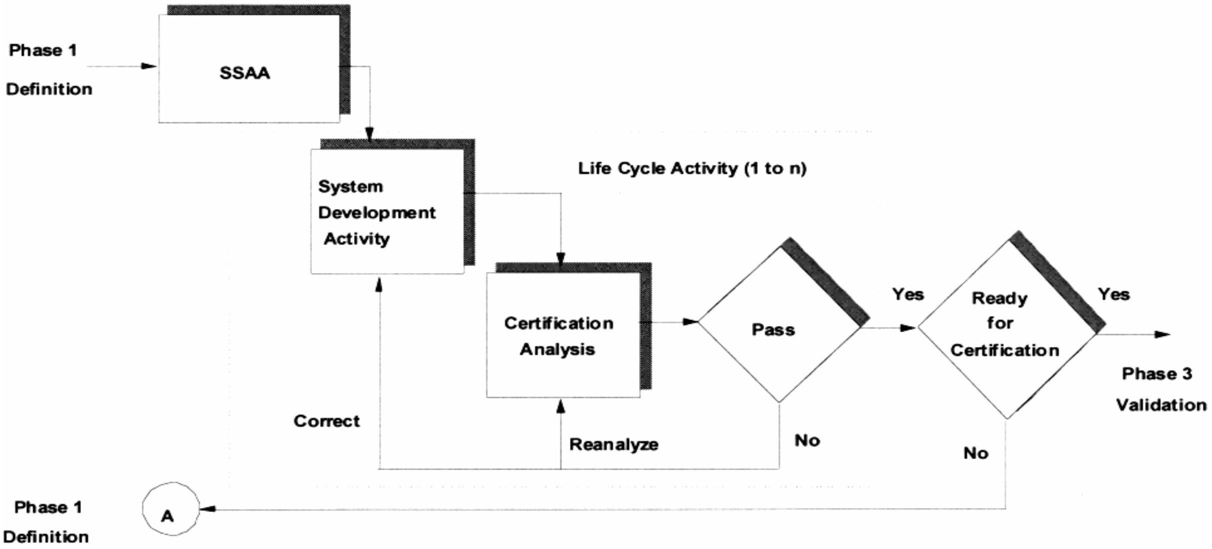

Figure 7. Phase 2 – Verification (DODI 5200.40)

Phase 3 will continue after all tasks in Phase 2 are complete. Phase 3 includes conducting tests to validate that the target system's security controls meet applicable requirements and produces evidence to assist the designated approving authority (DAA) in making an informed decision to grant approval to operate the system.

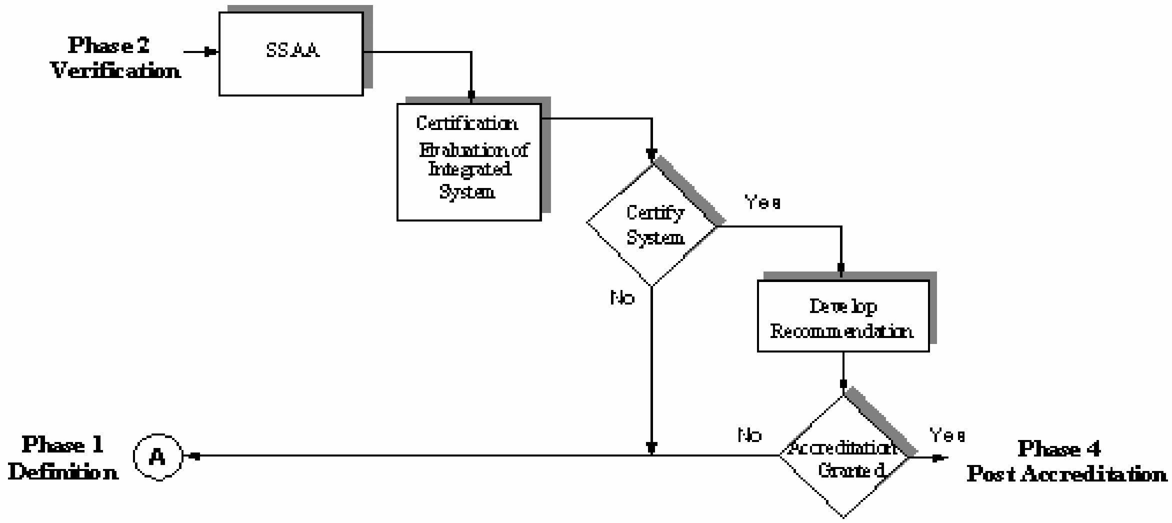

Figure 8. Phase 3 – Validation (DODI 5200.40)

The last phase of the process is called Post Accreditation. It begins after initial accreditation. The tasks will include monitoring changes to the operational and threat environments to ensure an acceptable level of residual risk is preserved and conducting periodic compliance validation reviews of security management and configuration management.

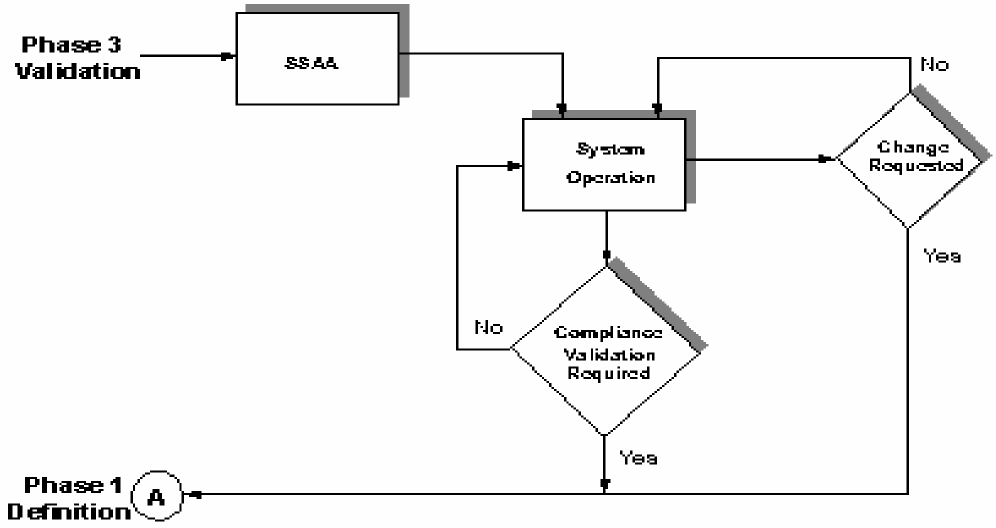

Figure 9. Phase 4 – Post Accreditation (DODI 5200.40)

DITSCAP is considered as the best process for DoD in the acquisition of buying COTS software DoD systems. DITSCAP ensures vulnerabilities are addressed and resolved. DITSCAP also enhances the security activities by tailoring to the system's lifecycle phase and program strategy. In addition, the process clearly defines detailed tasks for each phase. Each phase has entering and exiting criteria. Unfortunately, this is a very time consuming process but its quality is rewarding at the end of the process. Every risk or vulnerability will be reviewed by all stakeholders. The product that goes through DISCAP will have high confidence. The process also improves the life cycle of the product.

2. Recommendation for Software COTS Vendors

COTS software can be used in both DoD and industrial markets. COTS software has been implemented successfully in some DoD applications and other industrial applications. However, there are still cases where COTS fails in DoD and commercial applications. The lack of quality control, cost, and schedule driving the tendencies of the commercial world are the main causes of software failures. The current commercial producers' approach of "fail first, patch later" cannot fit within DoD requirements, especially for DoD mission-critical applications. The current approach also does not meet other commercial applications since the damages caused by software failure can put some business out of service. Currently, there is no law to punish software producers who sell bad software. However, in the future, it is expected that there will be a law to handle software makers based on damages they cause to other businesses who are the victims of their software. Commercial software vendors should have better methods to meet DoD security requirements and also to secure their products from the current increasing security threats. COTS software has a very low amount of error free code comparing to military software. For example, how frequent does Microsoft send out patches to fix security holes? And yet, Microsoft is considered the top software company worldwide. Many military software projects reach 95 percent in defect removal efficiency, and some have approached 99 percent for DoD weapons and communications systems (Caper). What makes military software have the high percentage of defect free code? DoD has succeeded in their software acquisition because they have a well-organized and formal quality control and formal project management process.

Commercial software producers should learn and adopt the success of the military process. Compared to industry, DoD is considered to have the best practice and most experience in dealing with secure software. DITSCAP has been a well-known process for DoD information system. Commercial software producers should also follow these DoD processes to certify their products due to the tremendous experience DoD has in security aspect. DITSCAP can be applied to either existing or newly developed commercial information technology products. Commercial software producers will benefit if applying DITSCAP since it secures the life cycle of a product. It also helps improve the commercial and corporation management approach. It keeps all stakeholders, from program manager to developers, users, and operators in the process to implement the product security functions. DITSCAP also ensures commercial employees are aware of their role to protect hackers and intruders from the corporation’s business as well as their product. The incorporation of DITSCAP will make security become a part of the company’s routine operation, not a nice-to-have feature.

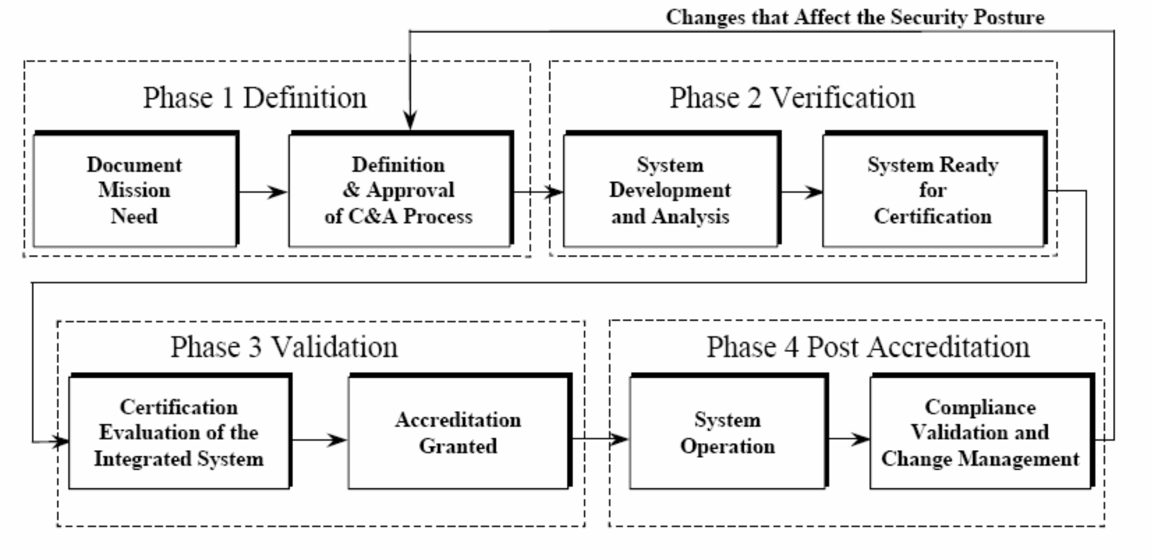
Changes that Affect the Security Posture
Phase 1 Definition
Document Mission Need
Definition & Approval of C&A Process
Phase 2 Verification
System Development and Analysis
System Ready for Certification
Phase 3 Validation
Certification Evaluation of the Integrated System
Accreditation Granted
Phase 4 Post Accreditation
System Operation
Compliance Validation and Change Management

Figure 10. Simplified DITSCAP for Commercial Software Vendor (Oar)

Oar has simplified DITSCAP in how it can be used for commercial software corporations, as shown in Figure 10 (Oar).

With a four-phase DITSCAP approach, software companies not only gain the security credibility, but also improve the life cycle management process. Threats continue to grow in the Information Technology field and if commercial companies do not develop the adequate process to improve the security of their products, they end up paying a high price due to lost productivity and revenue. This concept is the same as if a person wants to drive, they have to pass the driving tests and get licensed. If COTS producers want to build better software with high confidence in the current competitive market, they have to improve their process. DoD could help commercial developers achieve this by sharing best DoD practices

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

Traditional DoD acquisition is often over budget and behind schedule. Today, DoD faces the challenge of budget reductions. Meanwhile the demands of providing high performance, reliable, and secure products are increasing which require tremendous amounts of software upgrades and new developments. Traditional DoD acquisition does not fit in the present DoD budget and requirements environments. This will force DoD to change the acquisition processes to overcome the budget constraints and meet the demands for secure software at the same time. One of the approaches that DoD uses is to use COTS products to meet the cost and schedule constraints of DoD programs. However, the approach of selecting COTS software to develop and upgrade DoD systems has still driven projects over budget and produced schedule delays. DoD acquisition has seemed to underestimate the dependability and security risks of COTS software, and the nature of "fail first, patch later" mentality of COTS software vendors.

In the late 1990s, a lot of research and papers identified the pros and cons of applying COTS software in DoD applications. The biggest issue that commercial products face is the security issue. DoD has its own security requirement due to the nature of the operational environment and critical missions. The approach in which the COTS software fails, the consumer can sue or ask for a refund, does not work in DoD environment. If DoD systems fail, lives may be on the line and the nation's security might be jeopardized. However, the COTS approach is not totally without benefit. Reality has proven that military can meet readiness requirements by using COTS components. The question comes down on how to make sure the acquisition of using commercial software meets requirements at some expected level. DoD acquisition has to clearly identify the security requirements and carefully select the right product that not only meets the performance and readiness requirements but also is fully compliant with DoD security certifying process. DITSCAP is currently the best process to certify a system. Components should be tested in the environment they operate, not on a standalone basis. One should be reluctant to recommend the Common Criteria process. It is based on the analysis rather than the real test in the real environment. In addition to that, some commercial software producers receive certification at EAL4 but their products are still at high risk of vulnerability. Patches are sent out on monthly basis.

The traditional DoD software approach did not have to deal with security issues since it isolates itself from the rest of the world by limiting access physically and electronically. There are locked doors with limited access or guards that make it hard to get close to the software. Software built in house is guarded with encryption, private networks, or other electronic means on top of physical guards in place at facilities. The threat to breach software security is much harder. However, the COTS solution increases the risk of security break-in. DoD acquisition personnel should carefully evaluate the COTS solution before making any decision due to the nature mission of DoD systems. COTS components may conflict with DoD security constraints.

Regarding the commercial software industry, the traditional way of putting security responsibility on consumers and buyers does not fit in with current trends. DoD acquisition has been changed in the way of management and requirements. The security requirements are now included in DoD specifications and Request for Proposals (RFPs). This means that commercial software corporations who want to sell their products to DoD need to rethink their approach. Again, software defects are the causes of vulnerabilities and security breaches; therefore, commercial software companies should address security at the system level and also address it through out the life cycle of the product. Currently, there are no laws about the punishment of commercial software corporations for distributing flawed software. However, buyers can ask for a refund which results in lost revenues and productivity. Commercial software corporations should investigate developing appropriate management processes to handle security and reduce defects at the highest level. In the future commercial developers may very well obtain a positive return on their investment in following DoD process such as DIACAP/DISCAP if the commercial marketplace becomes increasingly sensitive to the need for a high level of security in their software systems. Security of products will be greatly enhanced if commercial software developers follow a disciplined process much like the DoD's. Although it is expensive and time consuming initially, following a process like DIACAP may bring a great return when corporations establish formal quality control methods.

It is not recommended that commercial corporations should go through CCEVS since the environment of the certification is not really the environment in which the software operates. It is a waste of time and effort to do so. CCEVS only provides the buyers with false confidence. Successful software will be found where there are good management and quality methods.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

BBC News. *Software Bug Bites US Military.* Online. 18 March 2003. http://news.bbc.co.uk/2/hi/technology/2860189.stm. August 2006

Browning, Graeme. *Hack Attack*. Online. 01 August 1997. http://www.govexec.com/features/0897s2.htm. Accessed August 2006

Budden, Timothy. *Decision Point: Will Using a COTS Component Help or Hinder Your DO-178B Certification Effort?* Crosstalk. Online. November 2003. http://www.stsc.hill.af.mil/crosstalk/2003/11/0311Budden.html. Accessed August 2006

Caper, Jones. *Defense Software Development in Evolution. CrossTalk.* Online. November 2002, http://www.stsc.hill.af.mil/Crosstalk/2002/11/jones.pdf#search=%22The%20defense%20software%20community%22. Accessed August 2006

Carhart, Andrew, Chan, Kim, and Hu, Art, *The Study of Critical Systems in Military & Commerce*, Research Paper, Computer Science Department, Stanford University, 10 June 2000.

Clark, Jeffrey. and Pradhan, Dhiraj. *Fault Injection: A Method for Validating Computer-System Dependability*. IEEE Software. Issue June 1995

Costello, Sam. *'Nimda,' 'Code Red' still Alive and Crawling*. CNN News. Online. 08 May 2002. http://archives.cnn.com/2002/TECH/internet/05/08/nimda.code.red.idg/. Accessed August 2006

Dean, J. Li., *Issues in Developing Security Wrapper Technology for COTS Software Products*. Institute for Information Technology. February 2002

Department of Defense, *Department of Defense Directive (DODD) 5000 - Defense Acquisition Guidebook*. November 2004.

Department of Defense, *Department of Defense Directive (DODD) 8500.1- Information Assurance*. 24 October 2002.

Department of Defense, *Department of Defense Instruction (DODI) 5000.2 - Mandatory Procedures for Major Defense Acquisition Programs (MDAPS) and Major Automated Information System (MAIS) Acquisition Programs*. March 1996.

Department of Defense, *Department of Defense Instruction (DODI) 5200.40 - DOD Information Technology Security Certification Accreditation Process (DITSCAP)*. 30 December 1997.

Department of Defense, *Information Technology Reform Act (Clinger-Cohen Act).* Online. February 1996. http://www.cio.gov/Documents/it_management_reform_act_Feb_1996.html Accessed August 2006

Department of Software Engineering, Mälardalen University, Sweden, Report: *Software Implemented Fault Injection Used for Software Evaluation*, by Andreas Johansson, July 2002.

Du, Wenliang., *Vulnerability Testing of Software System Using Fault Injection*. Dissertation. Purdue University. 6 April 1998

Eland, Ivan. 2002. *Can Pentagon Be Run Like a Business?* Issues in Science and Technology. Online. http://www.issues.org/18.3/eland.html. Accessed August 2006

General Services Administration (GSA). *Federal Acquisition Regulation 1997.* Part 12.

GAO Report. *Computer Attacks at Department of Defense Pose Increasing Risks*. Online. May 1996. http://www.gao.gov/archive/1996/ai96084.pdf

GAO Report. *Embedded Computer System -Defense Does Not Know How Much It Spends on Software*. July 1992

GAO Report. *GAO preaches on software development model*. Online. 05 May 2006. http://www.gcn.com/online/vol1_no1/40682-1.html. Accessed August 2006

GAO Report. *National Partnership Offers Benefits, but Faces Considerable Challenges*. Online. March 2006. http://www.gao.gov/new.items/d06392.pdf#search=%22Report%20GAO-06-392%22. August 2006

GAO Report. *Strong Management Practices Are Needed to Improve DOD's Software-Intensive Weapon Acquisitions*. Online. March 2004. http://www.gao.gov/new.items/d04393.pdf#search=%22DOD%20spent%20%2421%20billion%20on%20software%22. Accessed August 2006

Holmes, Allan., *Federal IT Flunks Out*. Online. May 2006. http://www.cio.com/archive/051506/federal_IT.html?page=6. August 2006

Krebs, Brian., *FBI Pursuing More Cyber-Crime Cases*. Washington Post. Online. 04 November 2004. http://www.washingtonpost.com/wp-dyn/articles/A25579-2004Nov4.html. Accessed August 2006

Lipson, Horward., Mead, Nancy., and Moore, Andrew., *Can We Ever Build Survivable Systems from COTS Components?* Software Engineer Institute. December 2001.

Main, Alec., *Application Security: Protecting the Soft Chewy Center*. CrossTalk. Online. October 2005. http://www.stsc.hill.af.mil/Crosstalk/2005/10/0510Main.html. August 2006

Minkiewicz, Arlene., *Security in a COTS-Based Software System*. CrossTalk. Online. November 2005. http://www.stsc.hill.af.mil/crosstalk/2005/11/0511Minkiewicz.html. August 2006

Oar, Gerald. and Jackson, Robert., *The Benefits of Applying DOD Information Technology Security Certification and Accreditation Process to Commercial Systems and Applications.* NIST. Online. 1998. http://csrc.nist.gov/nissc/1998/proceedings/paperE2.pdf#search=%22The%20Benefits%20of%20Applying%20DOD%20Information%20Technology%22. August 2006

Poulsen, Kevin., *Satellite Systems Hackable – Study*. Online._ October 2002 http://www.theregister.co.uk/2002/10/09/satellite_systems_hackable_study/. August 2006

Raines, Franklin., *Raines' Rules on Federal Information Systems Investments*. 25 October 1996. Online. http://www.balancedscorecard.org/bkgd/Raines_rules.html. August 2006

Slabodkin, Gregory., *FBI Suspects Two Teens in DOD Systems Attack*. Government Computer News (GCN). 09 March 1997 Issue. Online. http://www.gcn.com/print/16_6/32724-1.html

Voas, Jeffery., *Certification Software for High-Assurance Environments*. IEEE Computer Society. Volume 16, Issue 4, July – August 1999.

Wait, Patience., *Defense IT security can’t rest on COTS*. September 2004. Online. http://www.gcn.com/print/23_29/27422-1.html. August 2006

Zhong, Qun. and Edward, Nigel., *Security Control COTS Components*. IEEE Computer Society. June 1998. Volume 31, Issue 6.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California